

基于嵌入式微控制器的一种新型 GPIO 扩展键盘技术

赵鹏飞, 俞建定*

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 提出了一种利用单片机 GPIO 口结构、二极管单向导电性以及电容充放电原理进行按键扩展的方案, 详细叙述了其原理及整体设计, 并给出主要软件程序. 这种方案扩展了 8 个按键, 但仅占用了 MCU 的 2 个 GPIO 口, 大大节省了硬件资源. 经实际验证, 此按键扩展方案简单实用, 特别适用于引脚较少功能且比较强大的 MCU 开发系统.

关键词: IO 口; 按键; 单片机; 电容

中图分类号: TP344.2

文献标志码: A

文章编号: 1001-5132 (2013) 02-0059-05

随着社会及电子产业的发展, 嵌入式微控制器由于其高性能、高代码密度、小硅片面积而在消费类电子产品、智能家电、仪表仪器及工业控制等领域中得到了越来越广泛的应用. 对于一些人机交互频繁的嵌入式系统来说, 键盘作为一种普遍使用的人机交互途径, 由于其直观、简单而具有不可替代的作用. 在嵌入式系统设计中, 通常的方法是使用芯片 GPIO 口来进行按键扩展, 因此, 尽可能使用最少的 IO 口来扩展更多的按键, 可以有效节省单片机资源. 笔者结合实际开发经验对多种传统键盘扩展技术进行了探讨^[1-4], 并提出一种新型的 IO 键盘扩展方案.

利用 8 个 IO 口做出 4×4 键盘, 虽然相对图 1 来说缓解了 IO 口耗费问题, 但扩展数量还是比较有限. 当芯片的 I/O 口比较充裕时, 我们可以采用独立式键盘或者 M×N 矩阵式键盘; 但当 I/O 资源比较紧张时, 这种方法显然不能满足需求, 特别是对于一

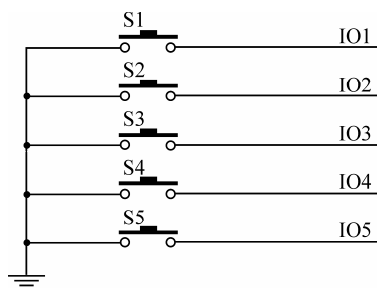


图1 独立式键盘原理图

1 传统 IO 扩展按键技术

1.1 独立式及矩阵式键盘

在项目开发中, 为了有效地实现人机交互(比如输入数据、命令等), 我们经常会用到比较多的按键. 传统的普通 IO 口扩展技术主要是独立式按键和矩阵式键盘式扩展

图 1 为独立式键盘原理图, 图 2 为传统矩阵式键盘原理图. 这 2 种键盘使用非常简单普遍. 独立式键盘最大的优势在于控制简单, 且可同时识别多个按键, 但缺点在于占用太多 IO 口, 如图 1 中的 5 个 IO 口只能有 5 个按键; 图 2 中传统矩阵式按键

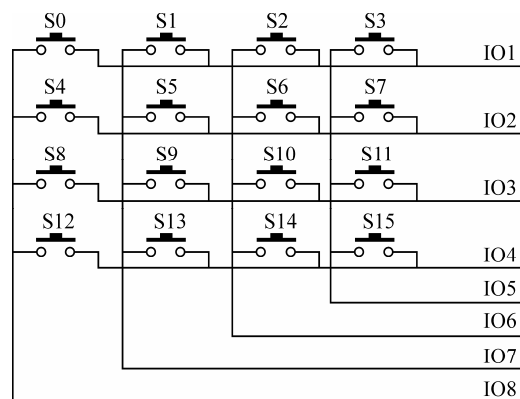


图2 传统矩阵式键盘原理图

收稿日期: 2012-10-19.

宁波大学学报(理工版) 网址: <http://nbdxlg.paperopen.com/>

第一作者: 赵鹏飞 (1985—), 男, 河南安阳人, 在读硕士研究生, 主要研究方向: 嵌入式系统. E-mail: zhpfnbu@sina.com

*通信作者: 俞建定 (1968—), 男, 浙江宁波人, 高级实验师, 主要研究方向: 嵌入式系统. E-mail: yujianding@nbu.edu.cn

些功能强大 I/O 口却比较少的芯片, 此类问题就更为突出.

1.2 利用单片机 AD 端口扩展键盘^[5]

系统开发中, 如果所选 IO 口支持 ADC 功能且有多余 A/D 转换通道的时候, 则在键盘扩展上可以利用 A/D 端口及分压电路扩展键盘^[6], 硬件电路原理如图 3 所示.

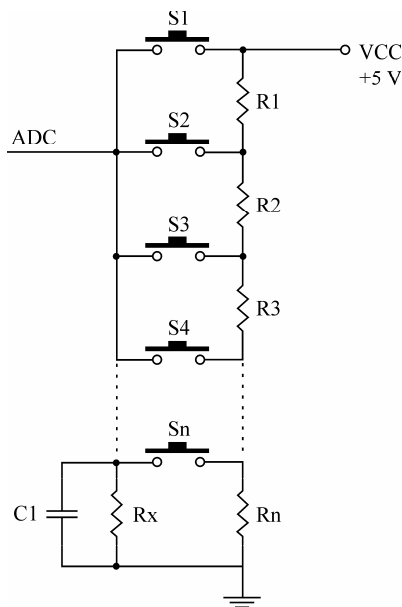


图3 利用 A/D 端口扩展键盘原理图

A/D 扩展必须要求用于扩展的端口具有电压检测功能, 且对 A/D 采样精度及电阻精度也有很高的要求, 因此, 显然普通 IO 口无法采用此种方法进行扩展, 很大程度上限制了此种扩展方法的应用范围.

1.3 其他 IO 扩展按键

随着电子技术的发展及嵌入式系统的应用范围的不断扩大, 也出现了一些其他 IO 扩展按键技术^[2-3], 但这些方案大多数仍然需要占用大量的 IO 口, 且电路和算法设计都比较复杂, 难以进入实际应用.

2 新型组合式按键扩展

我们知道 2 个普通 IO 口用传统方法进行按键扩展时, 只能做 2 个按键, 类似于图 1 中的形式 (2 个 IO 口无法利用传统方式进行组合, 所以只能是独立形式), 显然这对于系统应用的要求是远远不够的. 笔者在实际项目开发中, 结合多种 IO 扩展

按键技术, 开发出一种新式组合按键, 综合运用了二极管单向导电性、电容充放电原理、IO 输入输出模式转换等等, 用 2 个 IO 口实现了 8 个按键的扩展.

2.1 扩展硬件原理图

图 4 为新式组合键盘原理图, 左边 IO 口为 IO1, 右边 IO 口为 IO2, 下面横向排列 3 个按键分别为 S1、S2、S3, 中间纵向排列 3 个按键分别为 S4、S5、S6, 上面 2 个为 S7、S8, 最上面通过上拉电阻接 5 V 电源.

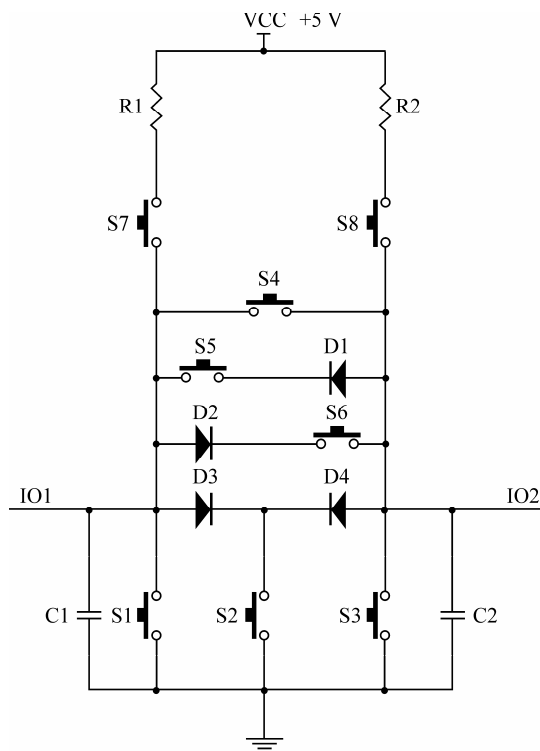


图4 新式组合键盘扩展原理图

2.2 原理及算法描述

2.2.1 S1、S2、S3 的检测

初始化 IO1 和 IO2 均输出状态为 1, 然后读取 2 个端口状态, 若 IO1 状态为 0, 而 IO2 状态为 1, 则可知有按键按下, 且是 S1 键; 若 IO1 状态为 1, 而 IO2 状态为 0, 则可知有按键按下, 且是 S3 键; 若 IO1 和 IO2 状态都为 0, 则可知有按键按下, 且是 S2. 这样, 就分别识别了 S1、S2、S3 按键.

2.2.2 S4、S5、S6 的检测

令 IO1 输出状态为 1, IO2 为 0, 读取 2 个端口状态, 若 IO1 电平跳变为 0, 则可知有按键按下, 且是 S4 或者 S6; 此时进入下一步判断, 交换 IO1

和 IO2 的状态, 令 IO2 为输出状态 1, IO1 为 0, 读取 2 个端口状态, 若 IO2 也跳变为 0, 则可知按下的按键一定是 S4, 若 IO2 保持 1 的状态不变, 由于二极管的单向导电性, 则可知按下的按键一定是 S6. 同样的方法, 可以识别出 S4 和 S5, 从而可以检测 S4、S5、S6 按键的情况

2.2.3 S7 和 S8 的检测

令 IO1 和 IO2 均输出低电平 0, 然后将 IO1 和 IO2 端口模式切换为仅输入模式, 此时若 S7 按下, 电源通过上拉电阻 R1 对电容 C1 充电, 经过一段时间后 C1 充电完毕, 此时 S7 松开, 电容将对地放电, 在一定的放电时间内读取 IO1 都将得到高电平, 所以可判断出按键 S7 是否被按下; 同理进行 S8 的检测, 检测完毕后 IO1 和 IO2 切换回准双向模式, 此时, 顺利完成 8 个按键的识别.

为保证 IO 口读取的状态不受前面输出状态的影响, 所以在 IO 口输出一定状态后, 都设置了短暂的延时.

2.3 软件设计流程图

扫描过程分为两步, 第一步首先扫描 S1、S2、S3、S4、S5、S6, 其算法流程如图 5 所示; 第二步则扫描 S7 和 S8, 其算法流程图如图 6 所示. 两个步骤可顺序执行, 互不影响.

需要特别说明的是在第二步的检测过程中, 利用了电容的充放电过程. 首先, IO1 和 IO2 为准双向模式, 输出均为 0, 作为端口清零操作, 然后将 IO1 和 IO2 设置为仅输入模式(Input only). 由于仅输入模式为高阻态, 在此种模式中仅有输入, 没有输出能力. 倘若 S7 和 S8 均未按下, 则此时读取 IO1 和 IO2 的状态一定是 0, 当 S7 按下时, 电源通过 R1 对 C1 充电, 这时读取 IO1 状态, 由于此时的 IO1 为高阻状态, 电容放电需要一个过程, 所以 IO1 状态为 1, 即可知按键 S7 按下. 同理可以识别按键 S8. 扫描完成后, 将 IO1 和 IO2 设置为准双向模式. 所以, 通过两步扫描过程可完成 8 个按键的扫描.

此时如果 IO 口处于准双向状态时, 由写变成读的状态时, 没有上拉电阻外接电源, 则有可能使读取到的状态都是 0, 从而导致无法准确读取端口的 1 状态. 此时就充分体现了电路中 2 个电容的重要作用, 在整个检测过程中均利用了电容的充放

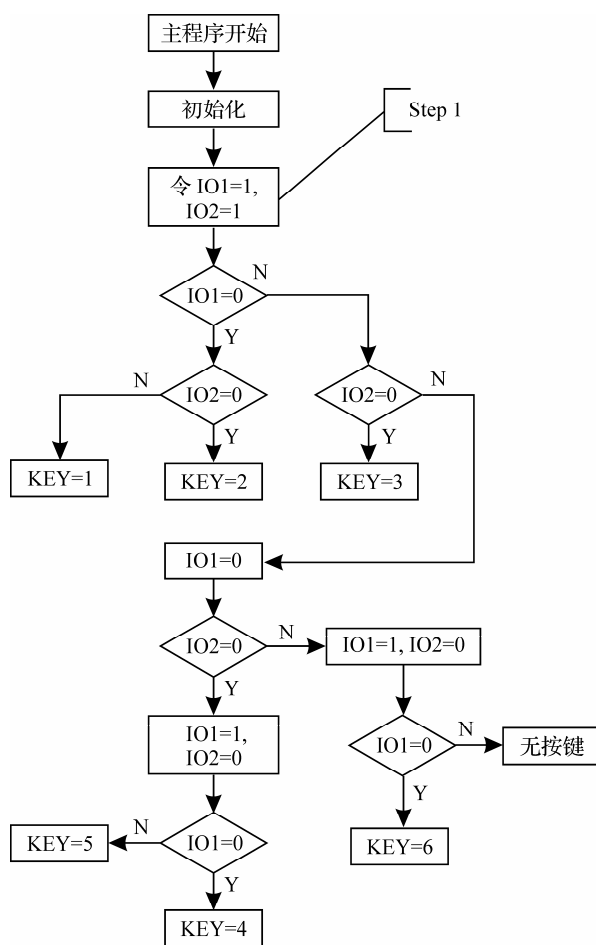


图 5 第一步扫描流程

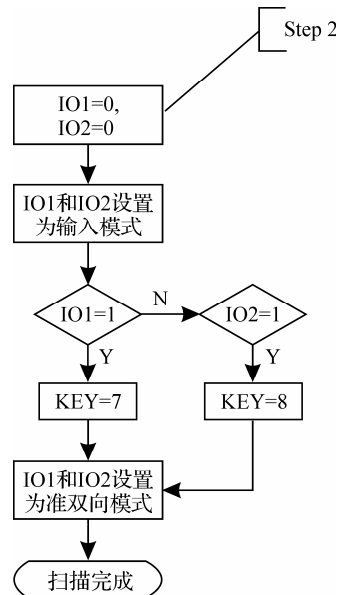


图 6 第二步扫描流程图

电原理. 例如, 当准双向端口为写状态 1 时, 可以对电容进行充电; 当读时, 由于电容存储的电能量

未经 IO 口拉低释放时,则相当于外接了高电平,足以让单片机进行判断。

值得思考的是,这种按键扫描过程需对 GPIO 口的结构有一个深入的探讨。笔者在测试过程中,扫描 S7 和 S8 时,如果还是准双向模式,此时若均初始化为 0,当转换为读状态时候,IO1 和 IO2 都马上变为 0,此时即使 S7 和 S8 按下,电源也不会对电容进行充电;若均初始化为 1,则开始时 IO1 和 IO2 对电容进行充电,由于电容存在一定的放电过程,此时无论 S7 和 S8 是否按下,读取端口状态则仍是 1。所以说,必须在读取之前进行 IO 口模式的转换。

电路中,器件选择 C1 和 C2 均为 103 μ F,电阻 R1、R2 均为 4.7 k Ω ,由示波器观察得知,充电时间大约为 1 ms,为防止状态之间的相互干扰,故在判断时候,需要作相应延时。

2.4 软件编写

由于篇幅限制,此处只给出扫描函数部分的关键代码;扫描过程也可利用定时器中断完成,在此就不再赘述。关键代码如下:

```
void scan_key(void)    //按键扫描
{
    IO1 = 1; IO2 = 1;
    delay_us(1000);    //延时
    if(!IO1)
    {
        if(!IO2)
            //tempkey = 3;    //读取键值
            P3 = 0x6d;
        else
            //tempkey = 2;
            P3 = 0x3d;
    }
    else
    {
        if(!IO2)
            //tempkey = 1;
            P3 = 0x48;
        else
        {
            IO1 = 0;
```

```
            delay_us(1000);
            if(!IO2)
            {
                IO1 = 1; IO2 = 0;
                delay_us(1000);
                if(!IO1)
                    //tempkey = 4;
                    P3 = 0x4b;
                else
                    //tempkey = 6;
                    P3 = 0x77;
            }
        else
        {
            IO1 = 1; IO2 = 0;
            delay_us(1000);
            if(!IO1)
                //tempkey = 5;
                P3 = 0x67;
            else
                //tempkey = 0;
                P3 = 0x00;
        }
    }
    IO1 = 0; IO2 = 0;
    delay_us(1000);    //延时
    P2CFG0 = 0x0a;    //IO 模式转换
    delay_us(1000);
    if(IO1 == 1)
        //tempkey = 7;
        P3 = 0x4e;
    else if(IO2 == 1)
        //tempkey = 8;
        P3 = 0x7f;
    else P3 = 0x00;
    P2CFG0 = 0x00;    //IO 模式转换
}
```

笔者在进行验证时按原理图搭建电路,并利用 1 个共阴极数码管接单片机 P3 口作为显示,程序经烧入芯片进行真实测试验证,按下不同按键

时, 数码管可以显示出不同数字, 程序运行稳定可靠。

3 结语

由于 S7 和 S8 的检测利用了电容充放电原理, 所以如果设置不同的电阻和电容值, 可以改变充放电时间, 然后再结合不同的延时时间, 按照扫描 S7 和 S8 的原理, 还可以扩展更多个按键。不过这样就需对电阻和电容的精度以及程序延时时间提出更高的要求。

按键是单片机及各种嵌入式系统中不可缺少的部分, 担负着人机交互、信息传递等重要的作用。阐述的按键扩展设计方案, 最大程度上节省了单片机的 IO 端口, 从而节省硬件资源, 降低产品投入成本, 并且软件编写简单清晰, 可靠性高。此方案在深圳赛元公司 SC91F72B 型单片机上已经顺

利实现, 对于需要节省 IO 进行按键扩展的项目具有一定的参考价值。

参考文献:

- [1] 黄菁, 刘青春. ARM 嵌入式系统 GPIO 扩展键盘设计[J]. 自动化应用, 2011(7):1-2.
- [2] 李运兵. 微控制器中按键处理技巧及应用[J]. 计算机应用, 2010, 19(2):195-197.
- [3] 王大庆. 扩展键盘电路及其编程. 国外电子元器件[J]. 2000(2):35-38.
- [4] 邹久朋, 张礼鸣, 林瑶瑶. 一种节省单片机 I/O 口的双矩阵键盘[J]. 单片机与嵌入式系统应用, 2008(3):70-71.
- [5] 方献良. 基于 A/D 的电容触摸按键电路设计[J]. 宁波大学学报: 理工版, 2010, 23(3):122-125.
- [6] 王曙光, 张宏升. 使用 ADC 作为单片机键盘接口[J]. 现代电子技术, 2004(9):62-63.

A New GPIO Extending Scheme Based on Embedded Microcontroller

ZHAO Peng-fei, YU Jian-ding*

(Faculty of Information Science and Technology, Ningbo University, Ningbo 315211, China)

Abstract: In this paper, a new interface extending scheme is proposed based on one-way electric conductivity of diode, capacitance charge-discharge principle and the structure of GPIO interface of MCU. Its principles and applications are described and the program is given. In this scheme extending 8 buttons only take 2 GPIO interface of MCU. Thus the hardware resources are saved greatly. Practice shows that this scheme is simple, practical and especially suitable for the MCU system which is powerful but has fewer pins.

Key words: I/O interface; button; MCU; capacitance

(责任编辑 章践立)